REMARKS and ARGUMENTS

Claims 1-36 are presented for examination; Claims 37-44 have been withdrawn in response to a restriction requirement.

The Office Action states at page 4, paragraph 7 that Claims 1-36 stand rejected under 35 U.S.C. 102(e) as being anticipated by Traversat. Although the Office Action has indicated that Traversat is a published patent application bearing publication number 2003/0200392, Applicants believe this to be a typographical error in the Office Action. In reality, such publication number is a published application to Wright et al. The Traversat and Wright references are cited correctly in the 892 form provided by the Examiner with the Office Action, but the names and numbers for these two references seem to have been reversed in the text of the Office Action. For purposes of responding to the Office Action, Applicants have assumed herein that the Examiner intended to base the prima face case of anticipation for Claims 1-36 on issued U.S. Pat. No. **6,957,237 to Traversal** and **not** on the published application to **Wright et al., 2003/0200392**.

Also, it is believed that another typographical error occurs in this statement at page 4, paragraph 7 of the Office Action – Applicants believe that, based on the rest of the Office Action, the Examiner meant to state that **only Claims 1-18 are rejected under 35 U.S.C. 102(e)** as being anticipated by Traversat, **not** Claims **1-36**. Applicants have fashioned their remarks below accordingly.

Claims 19-36 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Traversat in view of knowledge common in the art.

Claim rejections - 35 U.S.C. § 102(e)

Claims 1-36 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Traversat (U.S. Patent No. 6,957,237). However, a prima facie case of anticipation has not been made out regarding the claims. The rejection should be withdrawn and claims 1-36 should be allowed to issue.

Section 2133 of the MPEP recites (emphasis added): A claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

"[F]or anticipation under 35 U.S.C. 102, the reference must teach *every aspect* of the claimed invention ..." MPEP 706.02 (emphasis added). "The identical invention must be shown *in as complete detail as contained in the ... claim.*" *Richardson v., Suzuki Motor Co.*, 868 F. 2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added). Traversat simply fails to disclose every aspect of the claimed invention. The Examiner has therefore failed to meet his burden of making a prima facie case of anticipation and Claims 1-39 should therefore be allowed to issue.

Traversat relates to a mechanism to implement a database by caching portions of a virtual persistent heap into an in-memory heap for use by an application. (Traversat, Abstract). The Traversat heap includes a mixture of both code and data, with no indication that the code and data are segregated from each other in any way. See, e.g., Traversat at Col. 32, line 65 ("[a] heap may include code and data for use by the application.") The Traversat heap may also include structures for managing this mixed

code and data that is in the heap. See, e.g., Col. 4, lines 8-10. This mixed code and data that is stored in the Traversat heap may be portioned into pages or cache lines (Traversat, Col. 4, lines 9-11) and "at least some of the code and data in the heap for the application may be encapsulated into objects." (Traversat, Col. 4, lines 1-2).

As is pointed out by the Examiner in the Office Action, the **cache line** size of the portion of the Traversat heap that is cached in memory may be modified, (see Col. 29, lines 1-35). However, **the Traversat "cache line" is not a heap or compiled code cache**. Traversat states at Col. 29 that "[t]he term cache line may be used to refer to the smallest caching-in and caching-out granularity. A cache line corresponds to the smallest amount of data that can be loaded or flushed from the in-memory heap at one time." (Traversat, Col. 29, lines 2-6). There is no teaching, suggestion or disclosure in Traversat of any adjustment whatsoever to the size of the heap itself.

The Examiner also points out that, at Col. 8, lines 23 et seq., Traversat indicates that using garbage collection may be used to free up data in the heap that is no longer being used. However, Traversat does not disclose, suggest nor teach that the size of the heap may be adjusted; this section of Traversat simply indicates that space within the heap (which does not change size in terms of memory storage area allocated to it) for unused data and/or code may be made available for use. While this section teaches that the **amount of the data** itself may be variable, (See Traversat, Col. 8, line 21 – 24: "heap may include an area of ... memory... to store **data is some variable amount**", emphasis added), there is absolutely no teaching that the size of the heap itself, which is used to store this variable amount of data, is itself variable. Indeed, this section of Traversat indicates that in some cases, where the variable amount of data (and code) is

less than the size of the heap, unused portions of the heap may be available for use. Thus, the size of the heap is not adjusted if the variable amount of code and data to be stored in it is relatively small; this just results in portions of the heap being available for additional items to be stored in it. Thus, as is stated above, there is simply is no teaching, suggestion or disclosure in Traversat of any adjustment whatsoever to the size of the heap itself.

Claims 1, and 45. Claims 1 and 45 recite, in part, "making a first determination ... to indicate whether the size of a compiled code cache should be modified." Traversat does disclose at Col. 29 the increasing (lines 20-25) and decreasing (lines 30-35) of the **cache line** size. However, Traversat does not disclose, suggest or teach , "making a first determination ... to indicate whether the size of a compiled code cache should be modified." **Increasing or decreasing the size of a cache line does not read on, disclose, suggest nor teach modifying the size of a cache itself**. A prima facie case of anticipation therefore has not been made out regarding Claim 1 or Claim 45. Claims 1 and 45 are therefore allowable for at least the foregoing reasons. Claims 2-18, which depend from Claim 1, are also allowable for at least the foregoing reasons. Claims 46-52, which depend from Claim 45, are also allowable for at least the foregoing reasons.

Applicants disclose at paragraph 3 that "A heap is an area of memory reserved for the dynamic memory allocation needs of an application. It is thus reserved for data that is created at runtime ..." (Application, Page 2, lines 16-18 [Paragraph 3]).

Applicants disclose at paragraph 5 that "native instructions are stored in a compiled code cache." (Application, page 3, line 9 [Paragraph 5]). See also, page 6,

lines 12-13 [Paragraph 24]: "The native code of the method is stored in a compiled code cache 116 in memory."

Applicants disclose at paragraph 40 that "As is shown in Fig. 1, the heap 114 and the compiled code cache 116 occupy a single storage region 115." (Application, page 10, lines 22-23 [paragraph 40]). See also, paragraph 20: "a shared storage region 115 that includes both a heap 114 and compiled code cache 116. (Page 5, lines 13-14 [paragraph 20]). Although the compiled code cache and heap share a storage region, they are two separate and distinct memory areas: "the automatically managed runtime **memory areas include** a heap 114 and a compiled code cache 116." (Application, page 6, lines 4-5 [paragraph 23]) (emphasis added).

The foregoing cited portions of the Application, among others, provide support new Claim 45. Claim 45 recites, in part, "modifying a shared storage region, which is to store both the compiled code cache and the heap." As is explained above, **Traversat does not disclose, suggest nor teach a single storage area that is shared to store a heap memory area and a separate compiled code cache memory area**. Claim 45 is therefore allowable for at least this reason. Claims 46-52, which depend from Claim 46, are also allowable for at least this reason.

Claim 45 also recites, in part, "a heap, which is a memory area, separate from the compiled code cache, to store data created at runtime but that is not to store code." (Claim 45, in part). **Traversat does not disclose, suggest nor teach a heap memory area, separate from the compiled code cache, to store data created at runtime but that is not to store code.** Instead, Traversat expressly teaches at Col. 8, lines 24-27 that, although the Traversat heap may include data that is created at runtime, it "may also

include portions of code for the process." Claim 45 is therefore allowable for at least this reason. Claims 46-52, which depend from Claim 46, are also allowable for at least this reason.

Claim Rejections - 35 U.S.C. § 103(a)

Claims 19-36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Traversat in light of skill known in the art. However, the prima facie case of obviousness has not been made.

That is, there are elements of Claim 19 that are not disclosed, taught, nor suggested by Traversat nor by the knowledge of one skilled in the art. For such limitations, because they are not shown in either one, the combination of Traversat and knowledge of the skill in the art does not make out a prima facie case of obviousness.

For example, Claim 19 recites, in part, "making a first determination ... to indicate whether the size of a compiled code cache should be modified." Traversat does disclose at Col. 29 the increasing (lines 20-25) and decreasing (lines 30-35) of the **cache line** size. However, Traversat does not disclose, suggest or teach , "making a first determination ... to indicate whether the size of a compiled code cache should be modified." **Increasing or decreasing the size of a cache line does not read on, disclose, suggest nor teach modifying the size of a cache itself**.

Accordingly, a prima facie case of obviousness therefore has not been made out regarding Claim 19, and such claim should therefore be allowed to issue. Claims 20-36, which depend from Claim 19, are also allowable for at least the foregoing reasons.

Accordingly, Applicants respectfully submit that the applicable rejections have been overcome and must all be withdrawn. Applicants reserve all rights with respect to the application of the doctrine equivalents. Applicant respectfully requests that a timely Notice of Allowance be issued in this case. If the Examiner feels that an interview would help to resolve any remaining issues in the case, the Examiner is invited to contact Shireen Bacon of Intel, at (512) 732-3917.

Please charge any shortages and credit any overcharges to our Deposit Account No. 02-2666.

Respectfully submitted,

Dated: November 9, 2006

/Shireen Irani Bacon/
Shireen Irani Bacon
Registration No. 40,494